MONGO DB DATA STORAGE ARCHITECTURE

What is MongoDB?

MongoDB is one of the most powerful NoSQL databases and systems according to IntelliPaat (2021). It is an open-source document-based database management tool. It stores data in almost the same format as JSON. MongoDB is also very scalable, distributed, and flexible.

What is MongoDB Atlas?

MongoDB Atlas is a solution for cloud databases that are used for contemporary applications, it is also available globally (IntelliPaat, 2021). The automation allows for easy deployment of a managed MongoDb on Google Cloud, AWS, or Azure.

What About Rows And Columns?

MongoDB is a NoSQL tool, which means that it does not use rows and columns to associate data with each other, rather its **architecture** is built on collections and documents. A set of key-value pairs is used as basic units of data that allow documents to have different structures and fields.

The Data Model used

The model used in MongoDB is a highly elastic model that lets the user store and combine data of multivariate types. It also allows the user not to compromise on data access, powerful indexing options, or validation rules. When the user wants to modify schemas dynamically there is also no downtime, this implies that you can focus on making the data work harder instead of spending time on preparing data.

Why use MongoDB?

- No downtime while scaling
- Text search
- Graph processing
- Global replication

Advantages of using MongoDB

- Distributed Data Platform
- Fast and Iterative Development
- A Datamodel that is flexible
- Reduction of Total Cost of Ownership
- Integrated Feature Set
- Long term Commitment

Example of a document in MongoDB

```
{_id:
name: “Thomson”,
Age: 22,
Address: {{street: “124 church street”,
           city: “brooklyn”,
           state: “NY”,
           zip: “13400”,
           country: “US”}}
}
```

Mongos/Shards

There are tree components of a MongoDB sharded cluster:

- Shard
- Mongos
- Config servers

Shard: a shard in MongoDB is used to store the actual data. A shard provides high data consistency and availability of the database server. Each replica can be implemented to have a separate shard. A subset of the data will be contained by each of the shards in the database.

Mongos: When thinking of traditional databases a mongo in sharding will act as a query router. This means that it will give an interface between the client's application and the shared cluster. Mongos are used in order to process operations with shards then return the results to the user. More than one mongo instance can be divided to load a request.

Config servers: These config servers are used to store metadata of cluster servers. Metadata can contain information of cluster data set mappings. Mongos are uses the metadata to perform specific shards or operations. There can be more than one config server shared clusters in an environment.

Important MongoDB Facts:

Indexing: Any field in a document can be indexed

Replication: Native applications are used to maintain duplicate copies of data, thus preventing database downtime!

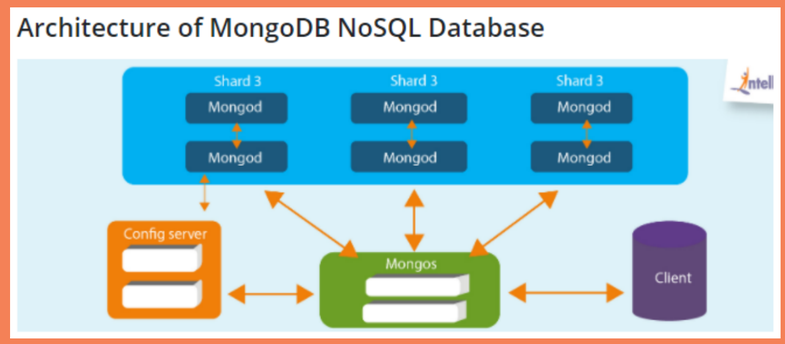
Auto-sharding: This is the distribution of data in multiple partitions and due to auto-sharding MongoDB does load balancing automatically.

MapReduce: MongoDB supports flexible aggrigation tools and MapReduce.

Schema-Less Database: As it is written in C++, it is a schema -less database.

Procedures: MongoDB JavaScript is mainly used as the database uses the language instead of procedures.

MongoDB Architecture



Database:

This is the physical container of your data and each container (database) has its own set of files on the file system. There can be multiple databases on a single MongoDB server.

Collection:

This is a group of database documents that are together. In RDBMS terms a collection is a table, and the entire collection exists in a single database instance. When it comes to collections there is no schema as inside of a collection different documents have different fields. Most of the time documents ina collectionhave the same purpose and serve the same goal.

Document:

A document is designed out of a set of key values and is associated with the dynamic schema. This benefits in the way that you that dynamic schemas allows the document that is in a single collection does not have to have the same fieldsor structures unlike a traditional database. Common fields alsodo not have to have the same data types.

Made by:

Enrico Dreyer 31210783
ITRI 623